A Web-Based Architecture for the Intelligent Management of Chronic Patients

Alberto Riva
Dipartimento di Informatica e Sistemistica, Università di Pavia, Pavia, Italy

*We describe a distributed architecture for medical informatics applications, based on the World-Wide Web (*WWW*) environment. After discussing previous experiences in the application of the* WWW *for medical purposes, we outline the features of a Common Lisp* HTTP *server designed to provide access to medical informatics applications using a standard Web browser. As an example of application, we describe a system for therapy planning and revision in the field of insulin-dependent diabetes. The system performs automatic data analysis and interpretation, and provides advice on possible adjustments to the therapeutic protocol that the patients are following, taking advantage of the network and multimedia capabilities offered by the* WWW *for user interaction.*

INTRODUCTION

From the point of view of information management, modern medical care can be described as a distributed collaborative activity. Care providers must cooperate to acquire patient data or to retrieve them from clinical data-bases, to interpret them and to plan an appropriate treatment. Moreover, the medical knowledge required to perform such tasks is necessarily distributed over different information sources and is represented using different formalisms. Improved communication, coordination and information sharing among the various agents involved in medical practice are recognized as being essential to the ability of making timely and informed decisions about the patient [6].

In general, no single and isolated computer-based system is able to perform the whole range of reasoning processes required by a complex medical problem. Although the diffusion of decision support systems for patient management is steadily increasing, their usefulness can be severely limited if they are conceived as *stand-alone* tools. On the contrary, they should be able to communicate with each other in order to exchange information, to use a variety of different knowledge representation formalisms and reasoning methodologies, and to interact with their users in a simple and effective way.

The above is especially true in the case of chronic diseases management. Chronic patients must be kept under strict control for a very large part of their life; chronic patients are preferably not hospitalized, and are often required to play an active role in the treatment of their disease. The system must therefore be *distributed* between the patient's house and the hospital, must rely on low-cost communication networks, and must be *usable* by a wide range of users, adapting its interface to their knowledge and skills.

In this paper, we will discuss how the use of the Internet and, in particular, of the World-Wide Web (WWW) infrastructure can provide a very effective and low-cost foundation for a distributed clinical information system. We will then describe a specialized HTTP server written in the Common Lisp language, that makes it possible to access applications over the Internet. Finally, we will show how we are exploiting this technology to build a distributed system for the intelligent management of patients affected by insulin-dependent diabetes mellitus.

BACKGROUND AND MOTIVATIONS

The WWW environment offers a set of facilities that can be extremely useful in the design of a hospital information system. The multimedia capabilities of most Web browsers allow the users to access knowledge and information stored under different forms. Web browsers are usually quite easy to use, and are therefore suitable for users categories, such as physicians and patients, that are not necessarily proficient in using computer systems. The use of the hypertext data structure as a high-level representation of the resources available in a distributed environment (such as a health care information system) makes it easy for the user to access them without having to be aware of their physical location.

The first experiences in the use of the WWW environment in a medical context have exploited its features to provide a friendly, powerful and consistent interface to traditional hospital information systems. Examples of applications include information sharing [1], clinical query management [5] and application development [4] in a medical environment.

Despite their success, these experiences have highlighted the limitations of the basic WWW paradigm when applied to tasks that are more complex than simple document distribution. In particular, the most common complaints regard the limitations in user input handling, the lack of a notion of *state*, the complexity of the CGI approach to application development [8].

A SPECIALIZED HTTP SERVER FOR DISTRIBUTED APPLICATIONS

LISPWEB is a full-featured HTTP server written in Common Lisp and designed to distribute *applications* instead of simple documents over the WWW [11]. The server handles GET and POST methods, forms, clickable images, dynamically generated images, and access control. Its main feature is that the pages it serves are not stored in "static" disk files, but can be generated by a Lisp function whenever they are requested.

The server provides the application developer with a mechanism to associate a URL with a Lisp function, and with a library of HTML-producing Lisp forms, that can be used in the function body. Whenever the user requests a URL associated with a function, the server runs it and sends the output, consisting of HTML code, to the client browser. This eliminates the need for external scripts, and of the associated CGI mechanism to pass arguments to the scripts and return the HTML page. The obvious advantages are faster page generation, a better integration between the application and the server environment, the ability to use the power and flexibility of the Lisp language to define the application interface in a simple and platform-independent way.

The LISPWEB development environment offers high-level interfaces to the different input-handling strategies that can be used in the WWW context. The most basic form of user input is the activation of a hyperlink to access a related document. Since the URLs pointed to by the hyperlinks are dynamically generated, they can be used to transmit additional information to the server. More in detail, when only a partial match can be found between the registered URLs and the requested one, the unmatched components of the request are passed to the Lisp function as arguments. For example, suppose the application contains a response function called `display-patient-data`, and associated with the */patient-data/* URL, that takes as arguments the patient identifier and the desired level of detail in the presented data. If the server were to receive a request for the */patient-data/129/high/* URL, it would call the `display-patient-data` function with "129" and "high" as arguments (of course, the function that generates the URL is responsible for providing the correct number of arguments). This technique can be used to hold information about the application state or the user's identity for authentication purposes; combined with access control, this can also be used to guarantee the security features needed when dealing with real patient data.

A second form of user interaction is based on *fill-in forms*: the user is presented with a page containing a number of *controls* (text fields, checkboxes, menus) and one or more *Submit* buttons, that are used to deliver the values of the controls to the server. LISPWEB handles forms by extracting the values of the controls from the header of the HTTP request and storing them inside a Lisp data structure. The values can then be passed as arguments to the response function or directly retrieved from the structure slots.

A more sophisticated form of user interaction is based on the capability to dynamically create and modify images. Combined with the ability to handle mouse clicks over images, this makes it possible to reproduce, within the limitations imposed by the nature of the HTTP protocol, the functionalities of a Graphical User Interface. The server generates an image representing the state of the interface, updates it by interpreting HTTP requests as user actions, and sends it back to the user as the response to the request.

The use of the LISPWEB server provides an effective solution to the *usability* and *accessibility* problems for new or existing Lisp applications, by taking advantage of the wide diffusion and flexibility of Web browsers [10]. Moreover, the server is able to communicate with external applications using an extension of the HTTP protocol; it can therefore become the heart of a distributed computing environment based on commonly available network technologies.

A WEB-BASED SYSTEM FOR DIABETES MANAGEMENT

In this section we will describe a computer system for the distributed management of insulin-dependent diabetic patients. The system is composed of a set of independent modules, that provide the different services needed in diabetes care.

Diabetes therapy

The common therapy of insulin-dependent diabetes mellitus uses exogenous insulin administrations to control the blood glucose level, trying to prevent dangerous oscillations. The diabetic patient plays an active role in the management of the disease, through the self-administration of the insulin injections according to a predefined protocol (that specifies the total insulin requirement, the number of daily injections, the insulin type and dose of each injection). The patient is also allowed to modify the insulin dosages in response to events such as an out-of-bounds level of blood glucose or a change in life-style, using simple adjustment rules provided by the physician along with the protocol. Finally, the patient must monitor and record the values of several physiological parameters, such as the blood glucose level (BGL) and glycosuria. The physician regularly evaluates the state of the patient during periodic visits during which the data gathered since the last visit are reviewed and, if needed, the therapeutic protocol is modified.

System architecture

Computer-based systems can be usefully applied to all the tasks that are typical of diabetes therapy. They can assist the patient in using the dose adjustment rules and in collecting and storing the measured data, thereby increasing the reliability of the collected data and possibly performing some preliminary analysis on them. They can assist the physician in the interpretation of the patient's data, and in criticizing and revising the therapeutic protocol through automated reasoning tools. They can handle the automatic transmission of the data from the patient's house to the clinic, thus increasing the frequency and the reliability of the communication.

Many examples of the application of computer systems to diabetes therapy exist in the literature [2, 7], although few of them were really successful. We believe that, in order to be effective, a computer-based system for diabetes therapy must be designed according to the constraints imposed by the characteristics of the disease and of the current therapeutic schemes. We have therefore defined a distributed architecture for therapy planning based on the cooperation among a set of services that deal with data management, information representation, and automated reasoning [9]. The system receives the monitoring data from a patient unit located at the patient's house, helps the physician in analyzing the data, and transmits the revised protocol back to the patient unit. We now briefly outline the modules that compose the current version of the system prototype. The LispWeb server is used to provide access to all the system functionalities.

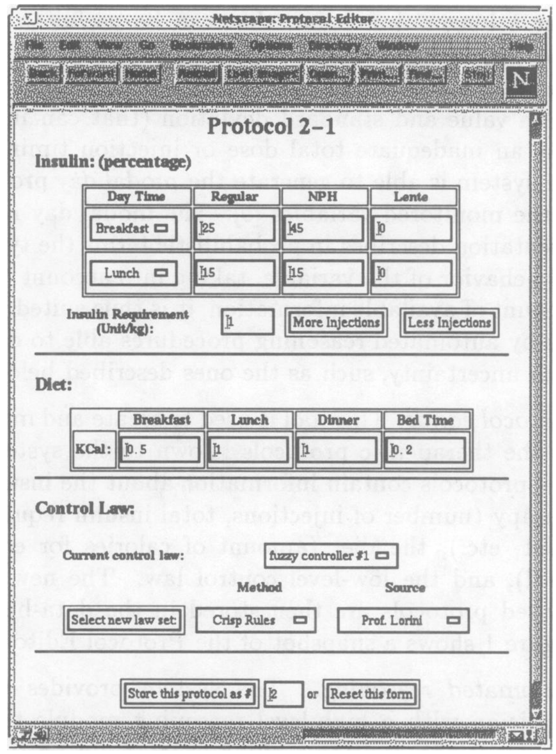

Figure 1: Protocol editing using a web-based graphical user interface.

Data-base. The system stores all patient data in a relational data-base. The relational data-base management system used in our prototype is able to receive queries and send the results back over the network. In this way, the data-base only acts as a repository of information, while the other modules take care of using such information or presenting it to the user.

Medical knowledge editor. The system's knowledge base comprises an explicit representation of the medical entities involved in diabetes therapy, and of the rules for abducing pathological conditions and for selecting suitable actions [12]. Both the ontology and the rules are represented through a *frame* system that is interfaced with the LispWeb server, allowing the user to inspect and modify the knowledge base entities using a Web browser.

Data analysis tools. There are basically two types of data analysis that the system can perform. The first type is applied the raw data as soon as they are collected by the patient; its goal is to identify short-term episodes and trends that can require an immediate adjustment to the insulin doses [3].

The second type of analysis aims at generating high-level descriptions of the state of the patient over a longer time span, that can be useful to revise the therapeutic protocol. In addition to the standard statistical indicators like the blood glucose mean value and standard deviation (that can indicate an inadequate total dose or injection timing), the system is able to generate the *modal day* profile of the monitored variables [9]. The modal day representation describes in probabilistic terms the typical behavior of the variable, taking into account the amount of available information; it is thus suited for use by automated reasoning procedures able to deal with uncertainty, such as the ones described below.

Protocol editor. This tool is used to create and modify the therapeutic protocols known to the system. The protocols contain information about the insulin therapy (number of injections, total insulin requirement, etc.), the diet (amount of calories for each meal), and the low-level control law. The new or revised protocols are then stored in the data-base. Figure 1 shows a snapshot of the Protocol Editor.

Automated reasoning. The system provides the physician with a high-level reasoning module that can assist the decision-making process. In particular, the system uses modal day information to determine the time-slices where problems are most likely to occur, and uses his knowledge on glucose metabolism and insulin dynamics to suggest modifications to the current therapeutic protocol. The suggestions, obtained through a combination of rule-based reasoning, uncertainty management techniques, and model-based simulations, are used to move within the space of known protocols attempting to solve the problems that were detected during the previous phase [9]. The system also tries to minimize the changes to the current protocol and the risk of adverse consequences.

An example

We will now present a brief example of how the system might assist the physician in a protocol revision task. We assume that the physician has connected to the LISPWEB server using a common Web browser. After supplying the appropriate identification information (e.g., a password), the physician is presented with a list of the diabetic patients being monitored, and notices that the system signals an alarm with respect to one of them. In particular, the analysis of the most recently uploaded monitoring data revealed three episodes of hypoglicemia during the preceding week, of which two occurred at breakfast time and one at dinner. The physician therefore decides to examine the situation, and, by clicking on the patient's name, causes the relevant historical information to be retrieved from the data-base

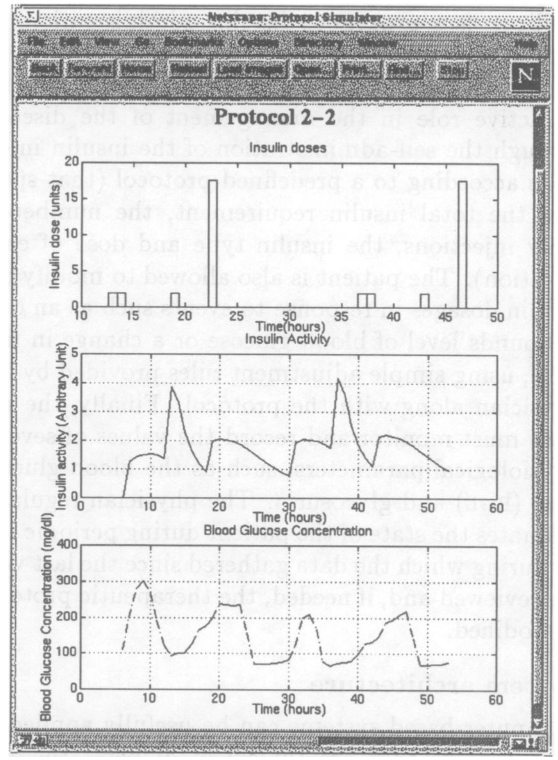

Figure 2: Insulin doses histogram and a simulation of glycemia and insulin activity profiles.

and converted into HTML form. Among other things (e.g. sex, age, weight, diabetes onset date, current protocol) the system displays the patient's current protocol, consisting in two injections of a mixture of regular and NPH (slow-acting) insulin, at breakfast and at dinner times.

The physician then visually inspects the monitoring data represented in graphical form, and invokes the automatic data analysis tools over the data. Although the average BGL value is normal, modal day analysis confirms that the probability of hypoglicemia at breakfast is too high, while no definite conclusion can be reached regarding dinner time due to the low amount of information available. Having decided that the current protocol needs to be modified, the physician requests the advice of the decision support system.

The outcome of the consultation is a revised protocol in which the NPH insulin dose at dinner has been reduced in favor of the regular insulin one, so that the number of injections and the total daily dose remain constant. The physician evaluates the new protocol through simulations of the resulting insulin activity and BGL profiles, such as the ones shown in Figure 2, and can perform adjustments to it using the protocol editor. When the physician finally confirms the changes, the new protocol is stored back in the data-base, to be transmitted to the patient the next time a connection takes place. The physician also includes a message asking the patient to provide more information about the hypoglicemic episode at dinner (e.g. the presence of exceptional physical activity), in order to help with its interpretation.

Note that in the course of this process the physician was able to access all the available tools using only a standard Web browser, instead of several different, specialized user interfaces.

CONCLUSIONS

The development of integrated hospital information systems, able to provide access to clinical data, medical knowledge, and decision support tools, is a long-standing problem. The WWW environment can potentially be very useful in this context, due to the simplicity of the HTTP protocol, the wide availability and ease of use of Web browsers, and the multimedia capabilities they offer.

We believe that the WWW environment can be exploited not only for simple document distribution and presentation, but also as the interface to a complete distributed computing environment. Applications developed using our Common Lisp HTTP server become "naturally" accessible over the Internet network, and are not tied to *ad-hoc*, incompatible user interfaces and communication protocols.

ACKNOWLEDGMENTS

The author sincerely thanks his advisor Mario Stefanelli, Riccardo Bellazzi, Marco Ramoni and Dr. Renata Lorini of the Pediatric Clinic of the University of Pavia for their collaboration and support.

REFERENCES

1. G. Almasi, A. Suvaiala, I. Muslea, C. Cascaval, T. Davis, and V. Jagannathan. Web* - a technology to make information available on the web. Technical report, CERC, West Virginia University, 1995. Submitted to the Fourth IEEE Workshop on Enabling Technology: Infrastructure for Collaborative Enterprises (WETICE 95).
2. S. Andreassen, J. Benn, R. Hovorka, K. G. Olesen, and E. R. Carson. A probabilistic approach to glucose prediction and insulin dose adjustment: description of metabolic model and pilot evaluation study. *Computer Methods and Programs in Biomedicine*, (41):153–165, 1994.
3. R. Bellazzi, C. Larizza, A. Riva, A. Mira, S. Fiocchi, and M. Stefanelli. Distributed intelligent data analysis in diabetic patient management. In J. Cimino, editor, *Proceedings of the 1996 AMIA Annual Fall Symposium*, 1996.
4. J. J. Cimino, S. A. Socratous, and P. D. Clayton. Internet as a clinical information system: application development using the World-Wide Web. *Journal of the American Medical Informatics Association*, 2(5):273–284, September 1995.
5. W. M. Detmer and E. H. Shortliffe. A model of clinical query management that supports integration of biomedical information over the world wide web. In *Nineteenth Annual Symposium on Computer Applications in Medical Care*, pages 898–902, New Orleans, LA, 1995.
6. V. Jagannathan, C. Gollapudy, R. Karinthi, K. Srinivas, R. Reddy, S. Reddy, and R. Shank. Architectural alternatives for community care networks. Technical Report CERC-TR-TM-94-001, CERC, West Virginia University, May 1994.
7. E. D. Lehmann, T. Deutsch, E. R. Carson, and P. H. Sonksen. AIDA: an interactive diabetes advisor. *Computer Methods and Programs in Biomedicine*, (41):184–203, 1994.
8. J. Rice, A. Farquhar, P. Piernot, and T. Gruber. Lessons learned using the web as an application interface. Technical Report KSL-95-69, Knowledge Systems Laboratory, Department of Computer Science, University of Stanford, September 1995.
9. A. Riva and R. Bellazzi. High level control strategies for diabetes therapy. In P. Barahona, M. Stefanelli, and J. Wyatt, editors, *Proceedings of the Fifth Conference on Artificial Intelligence in Medicine Europe*, number 934 in Lecture Notes in Computer Science, pages 185–196. Springer, June 1995.
10. A. Riva and R. Bellazzi. Current AI technologies for distributed patient management. In *Proceedings of the AAAI Spring Symposium on Artificial Intelligence in Medicine*, Stanford, CA, 1996. AAAI. (to appear).
11. A. Riva and M. Ramoni. Lispweb: a specialized HTTP server for distributed AI applications. *Computer Networks and ISDN Systems*, 28(711):953, 1996.
12. G. van Heijst, Th. Schreiber, G. Lanzola, and M. Stefanelli. Foundations for a methodology for medical kbs development. *Knowledge Acquisition*, 6:395–434, 1994.